UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/578,388	01/19/2007	David Spooner	042933/386642	1891

826   7590   10/27/2010
ALSTON & BIRD LLP
BANK OF AMERICA PLAZA
101 SOUTH TRYON STREET, SUITE 4000
CHARLOTTE, NC 28280-4000

EXAMINER
COYER, RYAN D

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
10/27/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/578,388	SPOONER ET AL.
	Examiner	Art Unit	
	Ryan D. Coyer	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS,
WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *1/19/2007*.

2a)☐ This action is **FINAL**.　　2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☒ Claim(s) *1-30* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 May 2006* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All　b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *5/5/2006; 10/20/2008*.

4)☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

This is in response to application 10/578388, filed on 1/19/2007. Claims 1-30 are

pending in the application, of which claims 1 and 27 are in independent form.

Drawings

New corrected drawings in compliance with 37 CFR 1.121(d) are required in this

application because <u>the drawings are unclear because they contain obscured, illegible

text</u>. Applicant is advised to employ the services of a competent patent draftsperson

outside the Office, as the U.S. Patent and Trademark Office no longer prepares new

drawings. The corrected drawings are required in reply to the Office action to avoid

abandonment of the application. The requirement for corrected drawings will not be held

in abeyance.

Specification

Applicant is reminded of the proper content of an abstract of the disclosure. A

patent abstract is a concise statement of the technical disclosure of the patent and

should include that which is new in the art to which the invention pertains. If the patent

is of a basic nature, the entire technical disclosure may be new in the art, and the

abstract should be directed to the entire disclosure. If the patent is in the nature of an

improvement in an old apparatus, process, product, or composition, the abstract should

include the technical disclosure of the improvement. In certain patents, particularly

those for compounds and compositions, wherein the process for making and/or the use

thereof are not obvious, the abstract should set forth a process for making and/or use thereof. If the new technical disclosure involves modifications or alternatives, the abstract should mention by way of example the preferred modification or alternative.

The abstract should not refer to purported merits or speculative applications of the invention and should not compare the invention with the prior art. (MPEP 608.01(b)). The abstract as filed asserts that "[b]ecause the elements execute under the control of a command line interface (and hence are command line programs) *it is far easier for a programmer to explore the functioning of the elements* - in particular how an element responds to a given input." (Emphasis added). This passage of the abstract refers to purported merits and speculative applications of the invention and compares the invention with the prior art. Applicant is directed to revise the abstract to comply with MPEP 608.01(b).

Claim Objections

Claims 1-30 are objected to because of various minor informalities. For example, claims 1, 2, 4, 5, 9, 17, 21, and 26 recite undefined acronyms. Claim 8 appears to have a minor typographical error in its third line, wherein it is recited "send instructions to the or each element" (sic), which Examiner believes should be corrected to read "send instructions to the or each an element." Claim 27 ends with a semicolon and a period but should end with only a period. All claims depending on or referencing the aforementioned claims have inherited the same deficiencies. Appropriate corrections are required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly
> claiming the subject matter which the applicant regards as his invention.

Claims 5-7 recites the limitation "the element under the control of a high level

interface." There is insufficient antecedent basis for these limitations in the claims

because the parent claim of each of the instant claims (claim 1) recites *plural* "elements"

under the control of a high level interface, precluding the control of only one single

element. In the interest of compact prosecution, Examiner will interpret the instant

limitations as if they were amended to read "an element under the control of a high level

interface". Appropriate corrections are required.

Claim 18 recites the limitation "[t]he method of Claim 1 in which the high level

language program can". There is insufficient antecedent basis for this limitation in the

claim because Claim 1 does not recite a "high level language program". However, the

context of the claim seems to indicate that Applicant intended for the instant claim to be

a dependent of claim 16. Accordingly, Examiner will interpret the claim as if it were

amended to read "[t]he method of Claim [[1]] 16". Appropriate correction is required.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of
> matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the
> conditions and requirements of this title.

Claims 27-30 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The "software application" recited in the instant claims reasonably can be interpreted as comprising nothing more than a series of software instructions. Moreover, the instant claims do not recite actual hardware implementation. Accordingly, the instant claims are rejected as being directed to non-statutory subject matter, namely software per se.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

Claims 1-7, 15, 19, and 22-30 are rejected under 35 U.S.C. 102(a) as being anticipated by WIPO International Publication Number WO 03/036470 A2, published 1 May 2003, hereinafter "Spooner."

Regarding claim 1, Spooner anticipates **"[m]ethod of rapid software application development for a wireless mobile device, the application being an enterprise networked application in which the device communicates with an enterprise server over one or more types of network connection;**[1] (see Spooner, pg. 2 ln. 15-17; "a new software development methodology that allows programs to be

---

[1] A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and

rapidly and efficiently developed for resource constrained mobile devices") **the method**

comprising the steps of:

     (a) a developer using a standard, high level interface (see Spooner, pg. 3 ln.

19-26; pg. 4 ln. 1-6, 15-20; "mStream" is a high level interface that "allows a small

(deliberately restricted), core library of basic building blocks or "primitives" to be built

once and re-used many times in different applications"; with mStream "very different

systems (and of arbitrary complexity) can be built using a small number of the above-

defined primitives") **(that is not specific to the mobile device OS)** (see Spooner, pg. 8

ln.15-16; mStream has been implemented on multiple mobile device operating systems

and therefore is not "specific" to any one mobile device OS) **on a computer remote**

from the mobile device to call, over one of the network connections, modular

software elements running on the device, (see Spooner, pg. 15-17, mStream

applications are built from various software "modules" comprising, *inter alia*,

"MStreamMan"; "MStreamClient"; "MStreamShell"; "MConsole"; "TcpListener"; etc; pg.

21 ln. 9-21; "the TcpListener module is used to listen on several TCP ports each

connected to a service for use by a Desktop PC . . . [which] then connects to the

services it wishes to use and then communicates with those services by reading/writing

service specific packets of binary data"; the modules are themselves build from smaller

components called "pipes" and "bundles") **the modular elements each (i)**

encapsulating functionality required by the wireless mobile device (see Spooner,

pg. 15-17; each module encapsulates a subset of functionality required by the wireless

---

Kropa v. Robie, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

mobile device, including network connectivity ("TcpListener") and command line

processing ("MStreamShell")) **and (ii) executing on the device, under the control of**

an interpreter of the high level interface; (see Spooner, pg. 11 ln. 17-21; the

"manager process" executes on the device and "[s]tarts, owns and monitors"

applications composed of the various modules; pg. 15, ln. 4-9; "the mStream

implementation consists of the following core modules . . . MStreamMan . . . is a 'server'

process that manages all the mStream objects in its process space. It also includes a

set of functions that clients [i.e.,other modules] can use to request operations from the

process manager")

  (b) the developer causing elements on the device to be combined using a

scripting engine running on the device; (see Spooner, pg. 10 ln. 20-33; "Tasks are

units of code that manipulate pipes and bundles. mStream defines APIs that specify [ ]

how these units of code must be written and must be invoked. It is the highest level of

abstraction of an operation"; a developer would combine software modules by writing a

"task" or "pipe processor" which comprises a "script (a text file that is interpreted at run

time by some other code and which allows the developer to alter the script file") **and**

  (c) the developer exploring how different elements respond to inputs by

repeating steps (a) and (b). (see Spooner, pg. 10 ln. 20-33; a developer alters the

script file to change the mix of elements; pg. 14 ln. 14-21; "plugins are pipe processors"

that "can be tested individually via a console"; pg. 17 ln. 5-10; the "MStreamEcho"

module has a "typical role" as "part of testing" tasks written by developers).

  <u>Regarding claim 2</u>, Spooner anticipates **"the method of Claim 1 in which one**

**or more modular elements encapsulate device networking functionality that
relates to connectivity over one or more of the following: GPRS, 2G cellular,
CDMA, WCDMA, Bluetooth, 802.11, infra-red, IP networking,** (see Spooner, pg. 16
ln. 23-29; "TcpListener" encapsulates networking functionality that relates to IP
networking connectivity) **dial up, modem; HSCSD and EDGE."**

Regarding claim 3, Spooner anticipates **"the method of Claim 1 in which one
or more of the modular software elements encapsulate general mobile device
functionality."** (see Spooner, pg. 20 ln. 20-29; mView is a mStream application that
comprises modular software elements that encapsulate general mobile device
functionality such as user input/output).

Regarding claim 4, Spooner anticipates **"the method of Claim 3 in which the
general mobile device functionality relates to one or more of the following: call
control and handling; PIM functionality; SIM functionality; remote control,
including screen scraping and faking key presses;** (see Spooner, pg. 20 ln. 20-29;
mView is a mStream application that comprises modular software elements that
encapsulate general mobile device functionality such as user input/output in the context
of remote control of a mobile device, including simulating "keyboard and mouse/pen
movements") **monitoring, including processes, threads, memory and settings; UI,
including creating an application where the screen elements are specified from a
script; telephony, including monitoring and making carts; file system, including
reading writing files and folders, monitoring for changes; database, including
structured storage, retrieval, searching and monitoring of arbitrary application**

data; device personalization, including ringtones, wallpaper and settings.

Regarding claim 5, Spooner anticipates **"the method of Claim 1 in which an element under the control of a high level interface is a TCPIP interface which allows other programs on the device to be run upon receipt of an incoming connection or to make outgoing connections from the device under control of other device based programs."** (see Spooner, pg. 16 ln. 31 – pg. 17 ln. 3; the module "MStreamTCP" allows outgoing TCPIP connections initiated by programs running on the mobile device).

Regarding claim 6, Spooner anticipates **"the method of Claim 1 in which an element under the control of the high level interface implements a remote command execution protocol.** (see Spooner, pg. 20 ln. 20-29; mView is a mStream application that permits "remote control behaviour").

Regarding claim 7, Spooner anticipates **"the method of Claim 1 in which an element under the control of the high level interface implements a scripting language that allows scripts to be written which use other programs on the device also controlled by a command line interface.** (see Spooner, pg. 16, ln. 1-8; the "MStreamShell" module is controlled by the mStream interface and reads scripts to obtain commands to invoke other programs on the mobile device).

Regarding claim 15, Spooner anticipates **"the method of Claim 1 in which the standard interface of a modular software element** (see Spooner, pg. 11 ln. 1-16) **is the name of the element,** (see Spooner, pg. 11 ln. 8-9) **a set of command line options,** (see Spooner, pg. 11 ln. 5) **two input streams** (see Spooner, pg. 11 ln. 6-7)

and two output streams." (see Spooner, pg. 11 ln. 8-9).

Regarding claim 19, Spooner anticipates **"the method of Claim I in which the
modular software elements insulate the application developer from the specifics
of the operating system of the device by requiring the application developer to
understand the type of functionality to be deployed and not the specific operating
specific code needed to implement that functionality using the operating
system."** (see Spooner, abstract, pg. 3 ln. 1-26; composing software at a "very high
level of abstraction" requires only an understanding of the desired functionality and
generally does not require knowledge of low-level implementation details).

Regarding claim 22, Spooner anticipates **"the method of Claim 1 in which
modular software elements can be chained together to build complex
functionality."** (see Spooner, pg. 20 ln. 20-29; mView is an example of complex
functionality built by combining multiple modular software elements).

Regarding claim 23, Spooner anticipates **"the method of Claim 1 in which the
modular software elements execute on the device in the context of an identity and
associated permissions."** (see Spooner, pg. 6 ln. 23-30; the manager process
"checks whether the requesting client session (as identified by the unique ID) has been
granted the necessary read/write rights by the application".)

Regarding claim 24, Spooner anticipates **"the method of claim 23 in which
there is an identity server with secure permissions that provides and controls the
identity and associated permissions."** (see Spooner, pg. 6 ln. 23-30; the manager
process is an equivalent of the claimed identity server because the manager process

enforces permissions according to identity information")

Regarding claim 25, Spooner anticipates **"the method of Claim 24 in which the identity server is located on the device."** (see Spooner, pg. 6 ln. 23-30; the manager process is located on the mobile device).

Regarding claim 26, Spooner anticipates **"the method of Claim 1 in which the modular software elements execute on a CPU of the mobile device."** (see Spooner, pg. 3 ln. 3-17; "mobile computing devices" in this context comprise CPUs and execute software programs created "for mobile computing devices" on said CPUs).

Regarding claim 27, Spooner anticipates **"A software application developed using the method of Claim 1,** (see the rejection of claim 1) **the application comprising modular software elements,** (see Spooner, pg. 15-17, mStream applications are built from various software "modules" comprising, *inter alia*, "MStreamMan"; "MStreamClient"; "MStreamShell"; "MConsole"; "TcpListener"; etc; pg. 21 ln. 9-21; "the TcpListener module is used to listen on several TCP ports each connected to a service for use by a Desktop PC . . . [which] then connects to the services it wishes to use and then communicates with those services by reading/writing service specific packets of binary data"; the modules are themselves build from smaller components called "pipes" and "bundles") **the modular elements each (i) encapsulating functionality required by a wireless mobile device and** (see Spooner, pg. 15-17; each module encapsulates a subset of functionality required by the wireless mobile device, including network connectivity ("TcpListener") and command line processing ("MStreamShell")) **(ii) executing on the device, under the control of**

an interpreter of a high level interface." (see Spooner, pg. 11 ln. 17-21; the "manager

process" executes on the device and "[s]tarts, owns and monitors" applications

composed of the various modules; pg. 15, ln. 4-9; "the mStream implementation

consists of the following core modules . . . MStreamMan . . . is a 'server' process that

manages all the mStream objects in its process space. It also includes a set of functions

that clients [i.e.,other modules] can use to request operations from the process

manager")

      Regarding claim 28, Spooner anticipates **"the software application of Claim 27

which is initiated or controlled from a remote application development

computer."** (see Spooner, pg. 21 ln. 9-21; "Symbian Connect" is an application initiated

and controlled from a remote computer).

      Regarding claim 29, Spooner anticipates **"the software application of Claim 28

which is accessed or controlled by the remote application development computer

in a secure fashion.** (see Spooner, pg. 21 ln. 9-21; Symbian Connect comprises the

use of a "secure sockets layer").

      Regarding claim 30, Spooner anticipates **"the software application of Claim 27

which runs stand-alone on the device without any initiation or control from a

remote application development computer."** (see Spooner, pg. 19 ln. 21 – pg. 20 ln.

18; the "MSurf" application runs as a 'stand-alone' application on the mobile device).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148

USPQ 459 (1966), that are applied for establishing a background for determining

obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 8-10, 12, 14, 16-18 and 20-21 are rejected under 35 U.S.C. 103(a) as

being unpatentable over Spooner in view of USPAT 6,389,560, hereinafter "Chew."

Regarding claim 8, Spooner anticipates **"the method of Claim 1"** and discloses

"an element on the device controlled by the high level interface" but does not

explicitly disclose the further claim limitations. Specifically, Spooner does not disclose a

instructions sent from a "high level language program" running on a remote computer

and that sends instructions to programs running on the mobile device. Spooner teaches

sending simulated button clicks, etc., from a remote computer, but does not explicitly

teach sending instructions to an application on the mobile device as claimed.

However, Chew discloses a method of controlling an application on a remote

device by sending individual commands in a "command line mode". (see Chew, col. 7

ln. 52 – col. 8 ln. 13). The command prompt executing on the local computer in Chew is

a high level language program that sends instructions to a program executing on a

remote device. Accordingly, Spooner in view of Chew obviates the further limitations **"in**

which a high level language program runs on an application development

computer remote from the device that can send instructions to an element on the

device" (see Chew, col. 7 ln. 52 – col. 8 ln. 13; "a user may input individual commands

which are interpreted and executed by the application . . . the command line mode also

allows the test application to be used remotely . . . the user may [ ] establish

communications with the test system and enter commands through the communication

link.").

       Chew and Spooner are both directed toward controlling the functionality of a

remote device and therefore are analogous art. The combination of Spooner and Chew

yields a method wherein command-line instructions are passed from a computer to an

application running on a mobile device that is remote from the computer, such that the

command-line instruction invokes some sort of functionality on mobile device. Such a

method obviates the instant claims. At the time of the invention, it would have appeared

obvious to one of ordinary skill in the art to combine Chew and Spooner as set forth

above because, as suggested in Chew, so doing would facilitate control of applications

running on remotely located devices. A clear benefit of combining Chew and Spooner

would have been the ability to remotely control modular applications running on a

remotely located mobile device. Accordingly, the instant claim is unpatentable over

Spooner in view of Chew.

Regarding claim 9, Spooner, in view of Chew, obviates **"the method of Claim 8
in which the high level language program is a command line program that
enables IP connections** (see Chew, col. 7 ln. 52 – col. 8 ln. 13; a "modem connection"
is an equivalent of the claimed IP connection) **between the mobile device and a
further program on the application development computer that implements the
same remote command execution protocol as the device."** (see Chew, col. 7 ln. 52
– col. 8 ln. 13; the computer application implements the same command line interface
that is used by the testing application on the remote device; see also Spooner, pg. 11
ln. 5; wherein tasks are invoked using command line entries. The combination of
Spooner and Chew yields a method wherein command-line instructions are passed
from a computer to an application running on a mobile device that is remote from the
computer, such that the command-line instruction invokes some sort of functionality on
mobile device.).

Regarding claim 10, Spooner, in view of Chew, obviates **"the method of Claim
9 in which rapid application development is achieved by enabling device
capabilities to be explored by executing the device-based elements controlled by
the high level interface from a command prompt** (see Spooner, pg. 11 ln. 5; pg. 16
ln. 1-14; "MStreamShell" is a command line processor that invokes other programs
according to received commands) **of the application development computer using
the remote command execution protocol."** (see Chew, col. 7 ln. 52 – col. 8 ln. 13;
Chew teaches that individual commands may be sent)

Regarding claim 12, Spooner, in view of Chew, obviates **"the method of Claim
9 in which rapid application development is achieved by using scripts which
combine the results of several device-based elements controlled by a command
line interface in the scripting language written on the device."** (see Spooner, pg. 11
ln. 24-26; tasks are combinations of several modules and may be implemented as
scripts; pg. 16 ln. 1-8; scripts are interpreted a command line entries).

Regarding claim 14, Spooner, in view of Chew, obviates **"the method of Claim
12 in which rapid application development is achieved by transferring the scripts
to the device and executing them, again using the computer command prompt.**
(see Spooner, pg. 11 ln. 24-26; tasks are combinations of several modules and may be
implemented as scripts; pg. 16 ln. 1-8; scripts are interpreted a command line entries;
the scripts would be sent to the mobile device prior to execution).

Regarding claim 16, Spooner, in view of Chew, obviates **"the method of Claim
8 in which the high level language is not restricted to a single type of high level
language, but can be any of the following depending on the requirements of the
developer of the software application: (a) a command line interface;** (see Chew,
col. 7 ln. 52 – col. 8 ln. 13; the computer application implements the same command
line interface that is used by the testing application on the remote device; see also
Spooner, pg. 11 ln. 5; wherein tasks are invoked using command line entries. The
combination of Spooner and Chew yields a method wherein command-line instructions
are passed from a computer to an application running on a mobile device that is remote
from the computer, such that the command-line instruction invokes some sort of

functionality on mobile device.). **(b) a scripting language;** (see Spooner, pg. 11 ln. 25;

Spooner teaches the use of scripts to define programs, the scripts would by definition

be written in a scripting language) **(c) a compiled language."** (see Spooner, pg. 11 ln.

25; "c++" and "Java" are compiled languages that can be used to define programs).

Regarding claim 17, Spooner, in view of Chew, obviates **"the method of Claim**

16 in which the application development computer is a desktop PC." (see

Spooner, fig. 10; "IBM Compatible" is a PC; Chew col. 8 ln. 5; a computer running Unix

would be an equivalent of a desktop PC).

Regarding claim 18, Spooner, in view of Chew, obviates **"the method of Claim**

16 in which the high level language program can in addition run on the device, to

enable re-programming of the device without the need to use a separate

application development computer." (see Spooner pg. 16 ln. 1-14; "MStreamShell"

and "MConsole" are both equivalents of the claimed "high level language programs" by

virtue of their command line interfaces; both run on the mobile device and can be used

on the mobile device to directly invoke programs).

Regarding claim 20, Spooner, in view of Chew, obviates **"the method of Claim**

8 in which the device runs a command interpreter (see Spooner, pg. 11 ln. 17-21;

the "manager process" executes on the device and "[s]tarts, owns and monitors"

applications composed of the various modules; pg. 15, ln. 4-9; "the mStream

implementation consists of the following core modules . . . MStreamMan . . . is a 'server'

process that manages all the mStream objects in its process space. It also includes a

set of functions that clients [i.e.,other modules] can use to request operations from the

process manager") **and the application development computer runs a command**

execution shell." (see Chew, col. 8 ln. 5; "Unix shell commands" disclose a command

execution shell).

Regarding claim 21, Spooner, in view of Chew, obviates **"the method of Claim**

8 in which the application development computer is connected to the device over

a local point to point IR, Bluetooth, USB, WAN, (see Chew, col. 7 ln. 52 – col. 8 ln.

13; a "modem connection" is an equivalent of the claimed WAN connection) **LAN,** (see

Chew, col. 7 ln. 52 – col. 8 ln. 13; a "modem connection" is an equivalent of the claimed

LAN connection; see also Spooner, fig. 10; TCP is used in a LAN) **SMS or GPRS or**

any combination of these.

Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Spooner

in view of USPGPUB 2004/0107385, hereinafter "Bates."

Regarding claim 11, Spooner, in view of Chew, obviates **"the method of Claim**

10" but does not disclose the further limitation **"in which an output of each command**

is shown at the command prompt on the application development computer."

However, Bates at para. 62 teaches that when a command is entered into a command

prompt, the results of that command are displayed in that command prompt. Bates,

Spooner, and Chew are directed toward computer testing and therefore are analogous

art. At the time of the invention, one of ordinary skill in the art would have found it

obvious to display the results of a command in the same command prompt where the

command was entered, as disclosed in Bates. A clear benefit of so doing would have

been the ability to efficiently monitor the results of entered commands. Accordingly, the

instant claim is unpatentable over Spooner, Chew, and Bates.

Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Spooner

in view of USPAT 6,249,886, hereinafter "Kalkunte."

Regarding claim 13, Spooner, in view of Chew, obviates **"the method of Claim**

12" but does not disclose the further limitation **"in which the script is composed in a**

text editor running on the application development computer." More specifically,

Spooner and Chew teach the use of scripts as claimed but do not explicitly teach that

those scripts may be composed in a text editor. However, composing scripts in a text

editor was very well known in the art at the time of the invention, as evidenced by

Kalkute at col. 6 ln. 1-8 ("The script file may be created manually using any standard

test editor). Kalkunte, Spooner, and Chew are directed toward computer testing and

therefore are analogous art. At the time of the invention, one of ordinary skill in the art

would have found it obvious to compose the scripts taught by Chew and Spooner in a

text editor, as taught by Kalkunte. A clear benefit of so doing would have been the

ability to compose the scripts on a wide variety of computing devices. Accordingly, the

instant claim is unpatentable over Spooner, Chew, and Kalkunte.

Conclusion

The prior art made of record on form PTO-892, 'Notice of References Cited', but not relied upon in the above rejections, is considered pertinent to applicant's disclosure. The aforementioned prior art addresses subject matter disclosed in the specification but not necessarily presented in the instant claims.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to RYAN D. COYER, whose telephone number is (571) 270-5306, and whose fax number is (571) 270-6306. The examiner normally may be reached via phone on Mon-Thurs, 9a-8p. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen, can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ryan D. Coyer/
Examiner, Art Unit 2191

/Wei Y Zhen/
Supervisory Patent Examiner, Art Unit 2191